# SentiKLUE: Updating a Polarity Classifier in 48 Hours

**Stefan Evert** and **Thomas Proisl** and **Paul Greiner** and **Besim Kabashi**
Friedrich-Alexander-Universität Erlangen-Nürnberg
Department Germanistik und Komparatistik
Professur für Korpuslinguistik
Bismarckstr. 6, 91054 Erlangen, Germany
{stefan.evert,thomas.proisl,paul.greiner,besim.kabashi}@fau.de

## Abstract

SentiKLUE is an update of the KLUE polarity classifier – which achieved good and robust results in SemEval-2013 with a simple feature set – implemented in 48 hours.

## 1 Introduction

The SemEval-2014 shared task on "Sentiment Analysis in Twitter" (Rosenthal et al., 2014) is a rerun of the corresponding shared task from SemEval-2013 (Nakov et al., 2013) with new test data. It focuses on polarity classification in computer-mediated communication such as Twitter, other micro-blogging services, and SMS. There are two subtasks: the goal of *Message Polarity Classification* (B) is to classify an entire SMS, tweet or other message as *positive* (pos), *negative* (neg) or *neutral* (ntr); in the subtask on *Contextual Polarity Disambiguation* (A), a single word or short phrase has to be classified in the context of the whole message.

The training data are the same as in SemEval-2013. The test data from 2013 are used as a development set in order to select features and tune machine learning algorithms, but may not be included in the training data. The 2014 test set comprises the development data, new Twitter messages, LiveJournal entries as out-of-domain data, and a small number of tweets containing sarcasm (see Rosenthal et al. (2014) for further details). For subtask B, there are 10,239 training items, 5,907 items in the development set, and 3,861 additional unseen items in the new test set. For subtask A, there are 9,505 training items, 6,769 items in the development set, and 3,912 additional items in the test set.

Our team participated in the SemEval-2013 shared task with a relatively simple, but robust system (KLUE) based on a maximum entropy classifier and a small set of features (Proisl et al., 2013). Despite its simplicity, KLUE performed very well in subtask B, ranking 5th out of 36 constrained systems on the Twitter data and 3rd out of 28 on the SMS data. Results for contextual polarity disambiguation (subtask A) were less encouraging, with rank 14 out of 21 constrained systems on the Twitter data and rank 12 out of 19 on the SMS data.

This paper describes our efforts to bring the KLUE system up to date within a period of 48 hours. The results obtained by the new SentiKLUE system are summarised in Table 1, showing that the update was successful. The ranking of the system has improved substantially in subtask A, making it one of the best-performing systems in the shared task. Rankings in subtask B are similar to those of the previous year, showing that SentiKLUE has kept up with recent developments. Moreover, differences to the best-performing systems are much smaller than in SemEval-2013.

## 2 Updating the KLUE polarity classifier

The KLUE polarity classifier is described in detail by Proisl et al. (2013). It used the following features as input for a maximum entropy classifier:

- The AFINN sentiment lexicon (Nielsen, 2011), which provides numeric polarity scores ranging from $-5$ to $+5$ for 2,476 English word forms, extended with distributionally similar words. For each input message, the number of positive and negative words as well as their average polarity score were computed.
- Emoticons and Internet slang expressions that were manually classified as positive, negative or neutral. Features were generated in the same way as for the sentiment lexicon.
- A bag-of-words representation that generates a separate feature for each word form that occurs in at least 5 different messages ($f \geq 5$). Only





*Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 551–555,
Dublin, Ireland, August 23-24, 2014.

| task | subset | rank | score | best |
|---|---|---|---|---|
| B | LJ14 | 3 / 42 | 73.99 | 74.84 |
| B | SMS13 | 4 / 42 | 67.40 | 70.28 |
| B | Twit13 | 6 / 42 | 69.06 | 72.12 |
| B | Twit14 | 10 / 42 | 67.02 | 70.96 |
| B | Sarcasm | 24 / 42 | 43.36 | 58.16 |
| A | LJ14 | 1 / 20 | 85.61 | 85.61 |
| A | SMS13 | 6 / 20 | 85.16 | 89.31 |
| A | Twit13 | 2 / 20 | 90.11 | 90.14 |
| A | Twit14 | 2 / 20 | 84.83 | 86.63 |
| A | Sarcasm | 2 / 20 | 79.32 | 82.75 |

Table 1: SentiKLUE results in SemEval 2014 Task 9 (among constrained systems). See Rosenthal et al. (2014) for further details and rankings including the unconstrained systems.

single words (unigrams) were used, since experiments with additional bigram features did not lead to a clear improvement.

- A negation heuristic, which inverts the polarity score of the first sentiment word within 4 tokens after a negation marker. In the bag-of-words representation, the next 3 tokens after a negation marker are prefixed with `not_`.
- For subtask A, these features were computed both for the marked word or phrase and for the rest of the message.

In order to improve the KLUE classifier, we drew inspiration from two other systems participating in the SemEval-2013 task: NRC-Canada (Mohammad et al., 2013), which won the task by a large margin over competing systems, and GU-MLT-LT (Günther and Furrer, 2013), which used similar features to our classifier, but obtained better results due to careful selection and tuning of the machine learning algorithm.

Mohammad et al. (2013) used a huge set of features, including several sentiment lexica (both manually and automatically created), word n-grams (up to 4-grams with low frequency threshold), character n-grams (3-grams to 5-grams), Twitter-derived word clusters and a negation heuristic similar to our approach. Features with the largest impact in subtask B were sentiment lexica (esp. large automatically generated word lists), word n-grams, character n-grams and the negation heuristic, in this order. NRC-Canada achieved F-scores of 68.46 (SMS) and 69.02 (Twitter) in task B, as well as 88.00 (SMS) and 88.93 (Twitter) in task A.

Günther and Furrer (2013) claim that state-of-the-art results can be obtained with a small feature set if a suitable machine learning algorithm is chosen. They used stochastic gradient descent (SGD) and tuned its parameters by grid search. GU-MLT-LT achieved scores of 62.15 (SMS) and 65.27 (Twitter) in task B, as well as 88.37 (SMS) and 85.19 (Twitter) in task A.

We therefore decided to make use of a wider range of sentiment lexica, extend the bag-of-words representation to bigrams, implement character n-gram features, and experiment with different machine learning algorithms, resulting in the SentiKLUE system described in the following section.

## 3 The SentiKLUE system

SentiKLUE is an improved version of the KLUE system and uses the same tokenisation, preprocessing and negation heuristics; see Proisl et al. (2013) for details. The features described below are used as input for a machine learning classifier that predicts the polarity categories *positive* (pos), *negative* (neg) or *neutral* (ntr). As in KLUE and GU-MLT-LT, the implementations of the Python library scikit-learn (Pedregosa et al., 2011)[1] are used. We tested four different learning algorithms: logistic regression (MaxEnt), stochastic gradient descent (SGD), linear SVM (LinSVM) and SVM with a RBF kernel (SVM). Parameters were tuned by grid search and the best-performing algorithm was chosen for each subtask. SentiKLUE makes use of the following features:

- Several sentiment lexica, which are treated as lists of positive and negative polarity words. Numerical scores are converted by setting appropriate cutoff thresholds. For each lexicon, we compute the number of positive and negative words occurring in a message as features, with separate counts for negated and non-negated contexts.
  - AFINN (Nielsen, 2011)[2]
  - Bing Liu lexicon (Hu and Liu, 2004)[3]
  - MPQA (Wilson et al., 2005)[4]
  - SentiWords (Guerini et al., 2013)[5]; we cre-

[1] http://scikit-learn.org/

[2] http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

[3] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

[4] http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

[5] http://hlt.fbk.eu/technologies/sentiwords

ated two word lists with score thresholds of 0.3 and 0.1
  - Sentiment140 (Mohammad et al., 2013)[6], which was compiled from a corpus of 1.6 million tweets for NRC-Canada; we created separate lists for normal words and hashtags with a score threshold of 1.0
  - NRC Hashtag Sentiment Lexicon (Mohammad et al., 2013)[7], which contains words that exhibit a strong statistical association (PMI score) to positive or negative hashtags, also compiled for NRC-Canada; again, we created separate lists for normal words and hashtags with a score threshold of 0.8
  - a manual extension including synonyms, antonyms and several word lists from online sources, compiled by the SNAP team (Schulze Wettendorf et al., 2014)
  - an automatic extension with distributionally similar words (DSM extension), using a strategy similar to Proisl et al. (2013)
- Word form unigrams and bigrams. After some experimentation, the document frequency threshold was set to $f \geq 5$ for subtask B and $f \geq 2$ for subtask A.
- In order to include information from character n-grams, we used a Perl implementation of n-gram language models (Evert, 2008) that has already been applied successfully to text categorization tasks (boilerplate detection in the CLEANEVAL 2007 competition). We trained three separate models on positive, negative and neutral messages. We selected a 5-gram model ($n = 5$) with strong smoothing ($q = 0.7$), which minimized cross-entropy on the training data (measured by cross-validation). For each message in the training and test data, three features were generated, specifying per-character cross-entropy for each of the three n-gram models.[8]
- Counts of positive and negative emoticons using the same lists as in the KLUE system.
- The same negation heuristic as in KLUE.[9]

For subtask A, we chose a simplistic strategy and computed the same set of features for the marked word or phrase instead of the entire message. In order to take context into account, the three class probabilities assigned to the complete message by a MaxEnt classifier were included as additional features. No other features describing the context of the marked expression were used.

Optionally, features were standardized and prior class weights (2× for *positive*, 4× for *negative*) were used in order to balance the predicted labels. The best-performing machine learning algorithms on the development set were MaxEnt for subtask B (L1 penalty, $C = 0.3$) and linear SVM for subtask A (L1 penalty, L2 loss, $C = 0.5$), as shown in Table 2.

## 4 Experiments and conclusion

In order to determine the importance of individual features, ablation experiments were carried out for both subtasks by deactivating one group of features at a time. Tables 3 and 4 show the resulting changes in the official criterion $F_{p/n}$ separately for each subset of the development and test sets, as well as micro-averaged across the full development set (DEV) and test set (GOLD). Rows are ordered by feature impact on the full gold standard. Positive values indicate that a feature group has a negative impact on classification quality: results are improved by omitting the features (which is often the case for the Sarcasm subset).

The most important features are bag-of-words unigrams and bigrams, closely followed by sentiment lexica. Training class weights had a strong positive impact in subtask B, but decreased performance in subtask A. In our official submission, they were only used for subtask B. Full-message polarity is the third most important feature in subtask A. Other features contributed relatively small individual effects, but were necessary to achieve state-of-the-art performance in combination. They are often specific to one of the subtasks or to a particular subset of the gold standard.

The bottom half of each table shows ablation results for individual sentiment lexica, with all other features active. Key resources are the standard lexica (AFINN, Liu, MPQA) as well as Twitter-specific lexica (Sentiment140, NRC Hashtag). Noisy word lists (DSM extension, SNAP, SentiWords) have a small or even a negative effect. Surprisingly, the standard lexica seem to give misleading cues on the Twitter 2014 subset (Table 3).

[6] http://www.umiacs.umd.edu/~saif/WebPages/Abstracts/NRC-SentimentAnalysis.htm

[7] ibid.

[8] Note that these features had to be generated by cross-validation on the training data to avoid catastrophic overfitting.

[9] The full list of negation markers is *not*, *don't*, *doesn't*, *won't*, *can't*, *mustn't*, *isn't*, *aren't*, *wasn't*, *weren't*, *couldn't*, *shouldn't*, *wouldn't*. To our surprise, including further negation markers such as *none*, *ain't* or *hasn't* led to a decrease in classification quality.

| | | CV | development set | | | | | | test set (gold standard) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| task | classifier | $F_{all}$ | $F_{pos}$ | $F_{neg}$ | $F_{ntr}$ | $F_{all}$ | $F_{p/n}$ | acc. | $F_{pos}$ | $F_{neg}$ | $F_{ntr}$ | $F_{all}$ | $F_{p/n}$ | acc. |
| B | MaxEnt | .727 | .724 | .651 | .772 | .735 | **.688** | **.734** | .731 | .650 | .750 | .726 | .691 | .725 |
| B | SGD | .725 | .728 | .645 | .773 | .736 | .686 | **.734** | .733 | .656 | .749 | .727 | **.695** | **.726** |
| B | LinSVM | .702 | .687 | .604 | .743 | .700 | .646 | .701 | .699 | .599 | .716 | .689 | .649 | .690 |
| B | SVM | .702 | .721 | .631 | .742 | .716 | .676 | .712 | .729 | .636 | .720 | .709 | .683 | .706 |
| A | MaxEnt | .864 | .890 | .872 | .179 | .849 | .881 | .863 | .893 | .853 | .171 | .841 | .873 | .856 |
| A | SGD | .864 | .889 | .867 | .223 | .849 | .878 | .860 | .891 | .847 | .188 | .839 | .869 | .852 |
| A | LinSVM | .860 | .892 | .876 | .064 | .847 | **.884** | **.865** | .895 | .856 | .064 | .838 | **.875** | **.857** |
| A | SVM | .855 | .890 | .873 | .024 | .842 | .881 | .862 | .892 | .853 | .014 | .832 | .872 | .854 |

Table 2: Performance of different machine learning algorithms on the training data (CV), development set and test set ($F_{all}$ = weighted average F-score; $F_{p/n}$ = official score; best results highlighted in bold font).

| **Task B** | SMS | Twitter | DEV | LJ14 | SMS13 | Twit13 | Twit14 | Sarcasm | GOLD |
|---|---|---|---|---|---|---|---|---|---|
| – bag of words | −.0837 | −.0322 | −.0502 | −.0344 | −.0807 | −.0316 | −.0335 | +.0511 | −.**0430** |
| – sentiment lexica | −.0445 | −.0354 | −.0389 | −.0690 | −.0422 | −.0372 | −.0092 | +.0750 | −.**0363** |
| – training weights | −.0033 | −.0413 | −.0266 | −.0275 | −.0077 | −.0482 | −.0204 | −.0342 | −.**0294** |
| – emoticons | −.0071 | −.0107 | −.0087 | −.0006 | −.0067 | −.0105 | +.0004 | +.0492 | −.**0048** |
| – bow bigrams | −.0074 | −.0005 | −.0035 | +.0010 | −.0105 | −.0012 | −.0096 | +.0956 | −.**0028** |
| – feature scaling | −.0027 | −.0010 | −.0014 | −.0021 | −.0030 | −.0026 | −.0004 | −.0034 | −.**0020** |
| – character n-grams | +.0029 | −.0068 | −.0033 | +.0012 | +.0040 | −.0044 | −.0056 | +.0056 | −.**0015** |
| – negation | −.0098 | +.0019 | −.0014 | −.0016 | −.0049 | +.0002 | −.0012 | +.0351 | −.**0002** |
| – bow $f \geq 2$ | +.0017 | +.0026 | +.0022 | +.0004 | +.0021 | −.0003 | +.0021 | +.0171 | +.**0013** |
| sentiment lexica: | | | | | | | | | |
| – standard lexica | −.0206 | −.0135 | −.0152 | −.0245 | −.0234 | −.0124 | +.0035 | +.0586 | −.**0124** |
| – Twitter lexica | −.0026 | +.0000 | −.0019 | −.0118 | −.0073 | −.0007 | −.0094 | +.0034 | −.**0066** |
| – SentiWords | −.0008 | −.0010 | −.0009 | −.0034 | −.0015 | −.0005 | −.0075 | +.0165 | −.**0017** |
| – hashtag lexica | −.0011 | +.0021 | +.0005 | −.0045 | −.0039 | +.0035 | +.0011 | −.0302 | −.**0005** |
| – DSM extension | +.0047 | −.0032 | −.0002 | −.0070 | +.0039 | +.0022 | −.0025 | +.0392 | +.**0002** |
| – manual extension | −.0008 | −.0018 | −.0011 | −.0015 | −.0019 | +.0000 | +.0041 | +.0361 | +.**0009** |
| only standard lexica | −.0124 | −.0119 | −.0120 | −.0088 | −.0101 | −.0108 | −.0095 | +.0439 | −.**0094** |
| only DSM extension | −.0303 | −.0260 | −.0262 | −.0427 | −.0287 | −.0251 | +.0021 | +.0183 | −.**0230** |

Table 3: Results of feature ablation experiments for subtask B. Values show change in $F_{p/n}$-score if feature is excluded. Rows are sorted by impact of features on the full SemEval-2014 test data (GOLD).

| **Task A** | SMS | Twitter | DEV | LJ14 | SMS13 | Twit13 | Twit14 | Sarcasm | GOLD |
|---|---|---|---|---|---|---|---|---|---|
| – bag of words | −.0283 | −.0252 | −.0256 | −.0207 | −.0292 | −.0249 | −.0411 | −.0041 | −.**0273** |
| – sentiment lexica | −.0027 | −.0231 | −.0151 | −.0078 | −.0023 | −.0245 | −.0144 | −.0109 | −.**0141** |
| – context (class probs) | +.0027 | −.0050 | −.0022 | −.0105 | +.0017 | −.0057 | −.0171 | +.0390 | −.**0062** |
| – negation | −.0081 | −.0041 | −.0052 | −.0064 | −.0063 | −.0024 | −.0058 | +.0000 | −.**0043** |
| – bow bigrams | −.0045 | −.0009 | −.0022 | −.0014 | −.0046 | +.0007 | −.0033 | +.0208 | −.**0014** |
| – character n-grams | −.0015 | +.0003 | −.0004 | +.0003 | −.0038 | +.0001 | −.0012 | +.0085 | −.**0012** |
| – feature scaling | +.0001 | +.0001 | +.0001 | +.0009 | +.0005 | −.0002 | −.0029 | −.0041 | −.**0004** |
| – emoticons | +.0023 | +.0026 | +.0025 | +.0016 | +.0038 | +.0012 | −.0062 | +.0000 | +.**0004** |
| bow $f \geq 5$ | +.0027 | +.0000 | +.0009 | +.0082 | +.0027 | +.0006 | −.0025 | +.0243 | +.**0015** |
| – training weights | +.0046 | +.0072 | +.0059 | +.0104 | +.0037 | +.0050 | +.0000 | −.0145 | +.**0040** |
| sentiment lexica: | | | | | | | | | |
| – standard lexica | −.0100 | −.0024 | −.0050 | +.0014 | −.0086 | −.0035 | −.0055 | +.0000 | −.**0044** |
| – Twitter lexica | −.0039 | −.0016 | −.0024 | −.0009 | −.0038 | −.0024 | −.0052 | −.0085 | −.**0031** |
| – hashtag lexica | −.0023 | −.0007 | −.0012 | +.0000 | −.0014 | −.0019 | −.0030 | −.0126 | −.**0017** |
| – manual extensions | −.0016 | +.0003 | −.0004 | +.0021 | −.0025 | −.0009 | +.0002 | +.0000 | −.**0007** |
| – SentiWords | +.0017 | +.0005 | +.0010 | +.0001 | +.0013 | −.0013 | +.0001 | +.0000 | −.**0002** |
| – DSM extensions | +.0099 | +.0011 | +.0044 | −.0008 | +.0098 | −.0006 | −.0004 | −.0085 | +.**0019** |
| only standard lexica | +.0030 | −.0038 | −.0011 | −.0019 | +.0035 | −.0048 | −.0027 | −.0168 | −.**0019** |
| only DSM lexica | −.0114 | −.0085 | −.0094 | −.0035 | −.0117 | −.0104 | −.0057 | −.0338 | −.**0089** |

Table 4: Results of feature ablation experiments for subtask A. Values show change in $F_{p/n}$-score if feature is excluded. Rows are sorted by impact of features on the full SemEval-2014 test data (GOLD).

## References


Stefan Evert. 2008. A lightweight and efficient tool for cleaning Web pages. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco.

Marco Guerini, Lorenzo Gatti, and Marco Turchi. 2013. Sentiment analysis: How to derive prior polarities from SentiWordNet. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1259–1269, Seattle, WA, October.

Tobias Günther and Lenz Furrer. 2013. GU-MLT-LT: Sentiment analysis of short messages using linguistic features and stochastic gradient descent. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 328–332, Atlanta, GA.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pages 168–177, Seattle, WA.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 321–327, Atlanta, GA.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 task 2: Sentiment analysis in Twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013)*.

Finn Årup Nielsen. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on Making Sense of Microposts: Big things come in small packages*, number 718 in CEUR Workshop Proceedings, pages 93–98, Heraklion, Greece, May.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Thomas Proisl, Paul Greiner, Stefan Evert, and Besim Kabashi. 2013. KLUE: Simple and robust methods for polarity classification. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 395–401, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment analysis in Twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, Dublin, Ireland.

Clemens Schulze Wettendorf, Robin Jegan, Allan Körner, Julia Zerche, Nataliia Plotnikova, Julian Moreth, Tamara Schertl, Verena Obermeyer, Susanne Streil, Tamara Willacker, and Stefan Evert. 2014. SNAP: A multi-stage XML pipeline for aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, Dublin, Ireland.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005)*, pages 347–354, Vancouver, BC, Canada.